

Tarea 7

Introducción y Taller de programación

I-Semestre 2022

Yarman Charpentier

Manfred Jones

Marco Rodríguez

26 de mayo de 2022

1. Sección 7.6.2 Árboles

1.1. Altura

1.1.1.

```
def altura(arb):  
    if arb == []: return -1  
    return 1+max(altura(arb[1]), altura(arb[2]))
```

1.2. Orden

1.2.1.

```
def preorden(arb):  
    if arb == []: return []  
    return [arb[0]] + preorden(arb[1]) + preorden(arb[2])
```

1.2.2.

```
def inorden(arb):  
    if arb == []: return []  
    return inorden(arb[1]) + [arb[0]] + inorden(arb[2])
```

1.2.3.

```
def postorden(arb):  
    if arb == []: return []  
    return postorden(arb[1]) + postorden(arb[2]) + [arb[0]]
```

1.3. Niveles

1.3.1.

```
def buscar_valor_principal(valor, arbol):
    niveles=[]

    def buscar_valor_auxiliar(valor, arbol, niveles, nivel):

        if arbol == []: return niveles
        if arbol[0]==valor:
            print('Entre')
            niveles.append(nivel)

        return buscar_valor_auxiliar(valor, arbol[1], niveles, nivel+1) + buscar_valor_auxiliar(valor, arbol[2], niveles, nivel+1)

    niveles=list(set(buscar_valor_auxiliar(valor, arbol, niveles, 0)))
    return niveles
```

1.3.2.

```
def nivel(k, arb):
    if arb == []: return []
    if k == 0: return [arb[0]]
    return nivel(k-1, arb[1]) + nivel(k-1, arb[2])
```

1.3.3.

```
def anchura(arb):
    anch = 0
    for niv in range(altura(arb)+1):
        anch = max(anch, len(nivel(niv, arb)))
    return anch

def nivel_mas_ancho(arb):
    mas_ancho=[]
    longitud_mas_ancho=0
    for niv in range(altura(arb) + 1):
        nivel_actual=nivel(niv, arb)
        longitud_nivel_actual=len(nivel_actual)
        if longitud_nivel_actual>longitud_mas_ancho:
            mas_ancho=nivel_actual
            longitud_mas_ancho=len(mas_ancho)

    return mas_ancho
```

1.4. Avanzados

1.4.1.

```
def nivelMax(arb):  
    if (arb == []): return -1  
    return 1+max(nivelMax(arb[1]), nivelMax(arb[2]))  
  
def nivelMin(arb):  
    if (arb == []): return -1  
    return 1 + min(nivelMin(arb[1]), nivelMin(arb[2]))  
  
def estaBalanceado(arb):  
    if arb == []: return True  
    return (nivelMax(arb)-nivelMin(arb))<=1
```

1.4.2.

1.4.3.

1.4.4.

```
def partirlista(lista, raiz):  
    contador=0  
    while (lista[contador]!=raiz):  
        contador+=1  
    return lista[:contador], lista[contador+1:]  
  
def reconstruirarbol(preorden, inorden):  
    if preorden != []:  
        raiz=preorden[0]
```

2. Bibliografía

- [1] Ramirez, E.(2017). Fundamentos de Programación. Sin editorial